# Model

- Input arrives in phases (online) and decisions taken must be monotone / consistent.
- competitive ratio (for minimization) = min over all input sequences $\left\{ \dfrac{\text{algorithmic solution}}{\text{(offline) optimal solution}} \right\}$

## E.g. ski-rental problem

- Unknown # of days $k$ that you will go skiing
- choice of buying skis for \$ B or renting every morning for \$ 1.
- OPT: If $k > B$, buy ; else, rent
- Online algorithm: rent for first B days, then buy.
- Competitive ratio = 2
- Is this the best?

## Online Lower bounds (Information theoretic)

- Algorithm and adversary play a game where they take alternate turns in deciding next output and input step respectively
- Consider the following adversary for ski-rental: stop skiing the moment skis are bought.

If algo buys skis before $k$ days, then

$$\left.\begin{aligned} OPT &= \min(k, B) \\ ALGO &= k + B \end{aligned}\right\} \frac{ALGO}{OPT} \geq 2.$$

Can randomization help?

Algo: Buy skis after $X$ days where $\Pr[X = x] = P_x$.

$$OPT = \min(k, B) = OPT(k)$$

$$E[ALGO] = \sum_{x=0}^{k} P_x (x + B) + k \sum_{x > k} P_x = ALGO_p(k)$$

We want $\min_{P_x} \max_{k} \frac{ALGO_p(k)}{OPT(k)}$.

At optimality, the above ratio must be invariant with $k$.
(Otherwise, we can shift probability mass to reduce max.)

Working in the continuous domain,

$$f(k) = \frac{\int_{x=0}^{k} (x+B) P_x \, dx + k \int_{x=k}^{\infty} P_x \, dx}{\min(k, B)}$$

First, consider $k > B$.

$$\frac{df}{dk} = \frac{1}{B} \left( (k+B) P_k - k P_x + \int_{x>k}^{\infty} P_x \, dx \right) = 0$$

$$\Rightarrow P_k + \int_{x=k}^{\infty} P_x \, dx = 0 \Rightarrow P_x = 0 \ \forall x > B.$$

Next, consider $k \leq B$.

$$\frac{df}{dk} = 0$$

$$\frac{df}{dk} = 0$$

$$\Rightarrow \quad ALGO_p(k) = \alpha \cdot OPT(k)$$

$$\Rightarrow \quad \frac{d\, ALGO_p(k)}{dk} = \alpha \cdot \frac{d\, OPT(k)}{dk}$$

$$\Rightarrow \quad (k+B)\,p_k - k\,p_k + \int_k^{\infty} p_x\, dx = \alpha$$

$$\Rightarrow \quad B\,p_k + \int_k^{B} p_x\, dx = \alpha$$

Taking derivative, $B \dfrac{d p_k}{dk} - p_k = 0$

$$\Rightarrow \quad p_k = A e^{k/B} \text{ for some constant } A$$

Now, $\displaystyle\int_0^B p_k\, dk = A \cdot B \cdot \int_0^B e^{k/B}\, d(k/B) = A \cdot B\,(e-1) = 1$

$$\Rightarrow \quad A = \frac{1}{B(e-1)}$$

Competitive ratio = Competitive ratio for $(k = B)$

$$= \frac{\dfrac{1}{B(e-1)} \displaystyle\int_0^B e^{x/B}(x+B)\, dx}{B}$$

$$= \frac{1}{e-1}\int_0^1 e^y (1+y)\, dy = \frac{1}{e-1}\left\{ e-1 + \int_0^1 e^y\, y\, dy \right\}$$

$$= 1 + \frac{1}{e-1} = \frac{e}{e-1}.$$

Exercise: Show that $\frac{e}{e-1}$ is tight for randomized algorithms.

## Paging/Caching

- Cache of size $k$
- Requests arrive online
- Objective: Minimize cache misses

Algo: Least Recently used (LRU):
- if requested page in cache, do nothing
- if requested page not in cache, evict page in cache that was used least recently

Lemma: The competitive ratio of LRU is $k$.

Proof: Consider maximal sequence of requests comprising $k$ pages. ALGO suffers $\leq k$ evictions and OPT suffers at least one eviction on the next request (for a distinct $(k+1)$ st page).

Lemma: There is a lower bound of $k$ for deterministic algos.

Proof: $k+1$ pages; adversary requests page outside cache. OPT evicts page that will be requested the latest in the future, i.e. at least $k$ requests later, whereas ALGO suffers cache miss for every request.

## Randomized paging (Marking)

Algo: in rounds

- each round starts with all locations unmarked
- when item in cache, mark the location
- when item not in cache, evict a random unmarked item and mark the location
- round ends when all locations are marked

<u>Analysis</u>: Consider an item "stale" if it was marked in previous but not in current one. If neither stale nor marked, call a location "clean".

Note that each round starts with all locations being stale and ends with all locations being marked.

Suppose, in a round, there are $l$ requests to clean items.

Let $d_I$ and $d_F$ be the size of the symmetric difference between the cache contents of OPT and ALGO at beginning and end of round.

OPT suffers cache miss for at least $d_F$ items that are not in ALGO's cache. Also, OPT suffers cache miss for at least $l - d_I$ items (since ALGO suffers cache miss for all requests to clean items).

Thus, $\text{OPT} \geq \max(l - d_I, d_F) \geq \dfrac{l - d_I + d_F}{2}$

Amortized over the round, $\text{OPT} \geq l/2$

ALGO suffers a miss for every clean item $\longrightarrow$ $l$ misses

For a stale item, the highest probability of suffering a miss is if all requests to clean items precede requests to stale items. For the $i$th request to a stale item, probability that it is a miss : $l$ (each of the $k-i+1$ stale items )

items. For the $i$ th request to a ...

that it is a miss is $\frac{l}{k-i+1}$ (each of the $k-i+1$ stale items has equal probability of having been replaced by a stale item)

Thus, expected # of cache misses $= \sum_{i=1}^{k-l} \frac{l}{k-i+1} + l = l \cdot H_k$.

## Randomized Lower Bound

Consider input comprising $(k+1)$ pages where the next page is chosen uniformly at random among all except the current page.

Divide into minimal segments with $k+1$ distinct pages and call these rounds. OPT misses only on last request of a round. How long is a round? Cover time of random walk on complete graph with $k+1$ vertices, which is $kH_k$.

ALGO misses every request with probability $1/k$

$\Rightarrow$ expected # of misses per round $= H_k$

Using Yao's minmax principle, we can conclude a lower bound of $H_k$ on the competitive ratio for randomized algorithms.